IN THE SPECIFICATION:

Please amend the specification as follows:

(1) The paragraph from page 3, line 8 to page 3, line 18 has been amended as follows:

In order to solve this problem, a common practice is to add ~~the use~~ a specialized "real time" operating system to allow the test system software to support accurate sequence control of the semiconductor test system (IC tester) and DUT ("Advantest T6682 Viewpoint architecture", Advantest, 1998). This generally runs on an additional processor(s). In some cases, such a real time operating system may also run on the same processor(s) as the general-purpose operating system. The use of the real time operating system usually allows timing resolution and repeatability that are between 100 to 1000 microseconds.

(2) The paragraph from page 3, line 19 to page 3, line 25 has been amended as follows:

However, the use of this additional operating system results in a configuration that has a non-homogeneous operating ~~systems~~ system running on multiple processors. This increases overall complexity in such as developing test programs and decreases flexibility in such as using application software. Consequently, this approach increases an overall test cost.

(3) The paragraph from page 7, line 21 to page 7, line 29 has been amended as follows:

An example of overall configuration of the semiconductor test system is shown in the block diagram of Figure 3 which includes the hardware and software components mentioned above. In the example of Figure 3, the semiconductor test system is configured by a tester hardware 28 and a power supply and tester peripherals 36. The tester hardware 28 includes a hardware control circuitry (pin electronics) 34 that allows high speed direct and indirect changes to ~~tester signal~~ test signals and power lines. As noted above with reference to Figures 1, for conducting a functional test, the tester hardware 28 in the semiconductor test system must be capable of producing the test signals (patterns) with the timing variability of 100 microseconds, which is not available when simply using a general-purpose operating system. By the configuration of Figure 3, it is possible to conduct a functional test of a semiconductor device under test with such a high timing resolution even when using a general-purpose operating system.

(4) The paragraph from page 10, line 21 to page 11, line 2 has been amended as follows:

Figure 6 shows an example of functional block diagram in the device driver 26 in the semiconductor test system of the present invention. The custom driver and hardware of the present invention are therefore designed in such a way ~~as~~ to allow the device driver 26 to make simple and fast decisions that directly and indirectly control the tester hardware 28. In this way, the device driver 26 is able to control the sequencing of the tester hardware 28, without violating restrictions that the operating system makes on the device driver 26. The device driver 26 is a privileged and reconfigurable software component that can be added to a general-purpose operating system in the host computer 22. The device driver 26 is provided to respond to the associated hardware in a timely fashion. In particular, the general purpose operating system (host computer 22) enables the device driver 26 to service hardware interrupts by allowing it to execute with a minimal time latency and with high priority.

(5) The paragraph from page 11, line 30 to page 12, line 4 has been amended as follows:

An example of operation in the semiconductor test system in the test start and end sequence is shown in the flow chart of Figure 7. As noted above, the test program is provided to the configuration software 32 from the host computer 22. Before the start of the test, at step 101, the configuration software 32 computes the initialization timing, power and signal configuration, deactivation timing, and etc. Based on the computation, the configuration software 32 creates consolidated test sequence data and sends the test sequence data to the device driver 26.

(6) The paragraph from page 13, line 7 to page 13, line 17 has been amended as follows:

In step 107, it is determined whether an interrupt is issued by the tester hardware 28. Such an interrupt is produced when the end of test logic 55 in Figure 5 generates an end of test signal. Typically, an end of test signal is generated when the response output of the DUT 40 is inconsistent with the expected data at the comparator 57. Upon receiving the interrupt based on the end of test signal, at step 108, the device driver 26 executes the deactivation sequence item to control the timings involving the end of the test. The device driver 26 stops executing the test pattern so that the test pattern ends at the specified timing.